Assignment 3 - Part B

Generated by Doxygen 1.12.0

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 2

Class Documentation

2.1 MonitoredSingleClientServer Class Reference

Public Member Functions

- MonitoredSingleClientServer (int port, SystemMonitor *monitor)

  Constructs a new Monitored Single Client Server.
- ~**MonitoredSingleClientServer** ()

  Destructor to clean up server resources.
- void run (int num_packets_to_process=-1)

  Runs the monitored single client server.

2.1.1 Constructor & Destructor Documentation

2.1.1.1 MonitoredSingleClientServer()

```
MonitoredSingleClientServer::MonitoredSingleClientServer (
            int port,
            SystemMonitor * monitor)  [inline]
```

Constructs a new Monitored Single Client Server.

Parameters

port	Port number to listen on
monitor	Pointer to SystemMonitor instance

2.1.2 Member Function Documentation

2.1.2.1 run()

```
void MonitoredSingleClientServer::run (
            int num_packets_to_process = -1)  [inline]
```

Runs the monitored single client server.

Parameters

num_packets_to_process	Number of packets to process (-1 for unlimited)

The documentation for this class was generated from the following file:

- server_TCP.cpp

2.2 PerformanceClient Class Reference

Performance client for TCP server.

Classes

- struct PerformanceMetrics
  Performance metrics.

Public Member Functions

- PerformanceClient (const std::string &server_ip, int port)
  Constructs a new PerformanceClient.
- PerformanceMetrics run_benchmark (int num_packets, int payload_size=1024)
  Runs performance benchmark.

Static Public Member Functions

- static void print_metrics (const PerformanceMetrics &metrics)
  Print performance metrics.

2.2.1 Detailed Description

Performance client for TCP server.

This class is used to run performance benchmarks on a TCP server.

2.2.2 Constructor & Destructor Documentation

2.2.2.1 PerformanceClient()

```
PerformanceClient::PerformanceClient (
            const std::string & server_ip,
            int port)  [inline], [explicit]
```

Constructs a new PerformanceClient.

Parameters

server_ip	Server IP address
port	Server port

2.2.3 Member Function Documentation

2.2.3.1 print_metrics()

```
static void PerformanceClient::print_metrics (
            const PerformanceMetrics & metrics)  [inline], [static]
```

Print performance metrics.

Parameters

metrics	Performance metrics to print

2.2.3.2 run_benchmark()

```
PerformanceMetrics PerformanceClient::run_benchmark (
            int num_packets,
            int payload_size = 1024)  [inline]
```

Runs performance benchmark.

Parameters

num_packets	Number of packets to send
payload_size	Size of each packet

Returns

PerformanceMetrics Calculated performance metrics

The documentation for this class was generated from the following file:

- client_TCP.cpp

2.3 PerformanceClient::PerformanceMetrics Struct Reference

Performance metrics.

Public Attributes

- double **total_time_ms**
- double **avg_latency_ms**
- double **total_data_sent_mb**
- double **bandwidth_mbps**

2.3.1 Detailed Description

Performance metrics.

The documentation for this struct was generated from the following file:

- client_TCP.cpp

2.4 SystemMonitor Class Reference

SystemMonitor class to monitor CPU and memory usage of the current process.

Public Member Functions

- void **start_monitoring** ()

  Start monitoring system resources.
- void stop_monitoring (uint64_t processed_packets)

  Stop monitoring system resources.

Static Public Member Functions

- static double **get_process_cpu_usage** ()

  Destructor to stop monitoring if still running.
- static long get_process_memory_usage ()

  Get current process memory usage.

2.4.1 Detailed Description

SystemMonitor class to monitor CPU and memory usage of the current process.

This class starts a separate thread to monitor CPU and memory usage of the current process. The monitoring thread reads CPU and memory usage every 100ms and stores the readings in vectors.

2.4.2 Member Function Documentation

2.4.2.1 get_process_memory_usage()

```
static long SystemMonitor::get_process_memory_usage ()  [inline], [static]
```

Get current process memory usage.

Returns

long Memory usage in KB

2.4.2.2 stop_monitoring()

```
void SystemMonitor::stop_monitoring (
            uint64_t processed_packets)  [inline]
```

Stop monitoring system resources.

Parameters

processed_packets	Total packets processed

The documentation for this class was generated from the following file:

- server_TCP.cpp

Index